PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Edwards	Examiner: D. Chung
Serial No.:	09/353,887	Group Art Unit: 2672
Filed:	July 15, 1999	Docket: 18195.29
Title:	GRAPHICS PROCESSOR WITH TEXTURE MEMORY ALLOCATION SYSTEM	

AMENDMENT AND RESPONSE AFTER FINAL OFFICE ACTION UNDER 37 CFR § 1.116

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant submits this Amendment and Response in response to the Office Action mailed on May 07, 2003.

Amendment to the claims begins on page 2.

Remarks begins on page 9.

AMENDMENT

1. (currently amended) A graphics accelerator for processing a graphical image, the graphics accelerator comprising:

a single texture buffer for storing texture maps and data relating to the texture maps stored in the texture buffer; and

a plurality of texture processors that perform texturing operations on the graphical image, the plurality of the texture processors retrieving texture packets from the single texture buffer,

each texture processor including a fetching engine that retrieves the texture packets, each texture packet being stored in the texture buffer and being associated with a texture map that is different than the texture maps associated with any other texture packet in the texture buffer, each texture packet including data relating to the location of its associated texture map in the texture buffer and data relating to the dimensional type of that texture packet's associated texture map.

2. (deleted)

3. (deleted)

4. (previously amended) The graphics accelerator as defined by claim 1 wherein the dimensional type of each texture map is one of a one dimensional texture map, a two dimensional texture map, and a three dimensional texture map.

5. (currently amended) The graphics accelerator as defined by claim 1 wherein the texture processor further includes:

an input for receiving a texture message indicating that a texture map is to be utilized by the texture processor, the fetching engine responsively retrieving selected texture packets from the single texture buffer in response to receipt of the texture message.

6. (original) The graphics accelerator as defined by claim 5 wherein the texture processor further includes:

a parsing engine for parsing a fetched texture packet and determining information relating to the texture map associated with the fetched texture packet.

7. (original) The graphics accelerator as defined by claim 6 wherein the information relates to the location in the texture buffer of the texture map associated with the fetched texture packet.

8. (original) The graphics accelerator as defined by claim 6 wherein the information relates to the number of dimensions of the texture map associated with the fetched texture packet.

9. (currently amended) A method of applying texture to a graphical image employing a graphics accelerator with a plurality of texture processors, the method comprising:

locating a texture packet identifying the location of a texture map in a single memory device, wherein the texture packet is associated with the texture map that is different than texture maps associated with other texture packets;

parsing the texture packet to determine the location of the texture map;

retrieving, based upon the determined location, the texture map from the single memory device; and

applying the texture map to the graphical image.

10. (original) The method as defined by claim 9 wherein the texture packet is located by accessing a record identifying the location of the texture packet.

11. (currently amended) The method as defined by claim 9 wherein the single memory device is texture memory.

12. (currently amended) The method as defined by claim 9 wherein the texture packet is stored in the single memory device

13. (currently amended) The method as defined by claim 9 further comprising

reconstructing the texture map after it is retrieved from the single memory device.

14. (original) The method as defined by claim 13 wherein the texture packet includes data relating to the dimensional type of the texture map, the texture map being reconstructed by parsing the texture packet to determine the dimensional type of the texture map, the texture map being reconstructed based upon the determined dimensional type of the texture map.

15. (currently amended) A computer program product for use on a computer system with a plurality of texture processors for applying texture to a graphical image, the computer program product comprising a computer usable medium having computer readable program code thereon, the computer readable program code including:

program code for locating a texture packet identifying the location of a texture map in a single memory device, wherein the texture packet is associated with the texture map that is different than texture maps associated with other texture packets;

program code for parsing the texture packet to determine the location and the number of dimensions of the texture map;

program code for retrieving, based upon the determined location, the texture map from the memory device; and

program code for applying the texture map to the graphical image.

16. (original) The computer program product as defined by claim 15 wherein the program code for locating includes program code for accessing a record identifying the location of the texture packet.

17. (currently amended) The computer program product as defined by claim 15 wherein the single memory device is texture memory.

18. (currently amended) The computer program product as defined by claim 15 wherein the texture packet is stored in the single memory device

19. (currently amended) The computer program product as defined by claim 15 further comprising:

program code for reconstructing the texture map after it is retrieved from the single memory device.

20. (original) The computer program product as defined by claim 19 wherein the texture packet includes data relating to the dimensional type of the texture map, the program code for reconstructing comprising:

program code for parsing the texture packet to determine the dimensional type of the texture map, the texture map being reconstructed based upon the determined dimensional type of the texture map

21. (currently amended) A graphics accelerator for processing a graphical image, the graphics accelerator comprising:

a single texture buffer for storing texture maps and data relating to the texture maps stored in the texture buffer; and

a plurality of texture ~~processor~~ processors that performs texturing operations on the graphical image, the plurality of the texture processors retrieving texture packets from the single texture buffer, ~~the~~ each texture processor including a fetching engine that retrieves texture packets, each texture packet being stored in the texture buffer and being associated with a texture map that is different than the texture maps associated with any other texture packet in the texture buffer, each texture packet including data relating to the dimensional type of its associated texture map.

22. (currently amended) The graphics accelerator as defined by claim 21 wherein each texture packet includes data relating to the location of its associated texture map in the single texture buffer.

23. (deleted)

24. (original) The graphics accelerator as defined by claim 21 wherein the texture processor further includes:

an input for receiving a texture message indicating that a texture map is to be utilized by the texture processor, the fetching engine retrieving selected texture packets from the texture buffer in response to receipt of the texture message.

25. (original) The graphics accelerator as defined by claim 24 wherein the texture processor further includes:

a parsing engine that parses a fetched texture packet and determines information relating to the texture map associated with the fetched texture packet.

26. (currently amended) A method of storing a texture map in a single linear texture memory of a graphics accelerator, the method comprising:

A. determining the dimension of the texture map;

B. converting the texture map to a one dimensional texture map if the dimension of the texture map is determined to be more than one dimensional, the one dimensional texture map having a first number of consecutive data blocks;

C. locating a second number of consecutive memory locations in the single texture memory, the first lumber being equal to the second number; and

D. storing the one dimensional texture map in the located memory locations in the single textured memory.

27. (original) The method as defined by claim 26 wherein the texture map is two dimensional, step B comprising:

B1. defining a plurality of data blocks within the texture map; and

B2. assigning a sequence number to each of the data blocks, the sequence numbers being consecutive numbers.

28. (original) The method as defined by claim 26 wherein step D comprises:

Dl. consecutively storing each consecutive data block of the one dimensional texture map in the located memory locations.

29. (currently amended) A graphics accelerator for processing graphical request code, the graphics accelerator comprising:

a single linear texture memory for storing texture maps;

a plurality of texture ~~processor~~ processors that applies textures to items to be displayed, the plurality of the texture processors retrieving texture packets from the single texture memory, ~~the~~ each texture processor including a texture map converter that converts texture

maps having dimensions greater than one dimensional to a one dimensional texture map, each dimensional texture map having a first number of consecutive data blocks, the texture processor further including means for locating a second number of consecutive memory locations in the texture memory, the first number being equal to the second number; and

means for storing the one dimensional texture map in the located memory locations in the single texture memory.

30. (original) The graphics accelerator as defined by claim 29 wherein the texture map converter comprises:

means for defining a plurality of data blocks within the texture map; and

means for assigning a sequence number to each of the data blocks, the sequence numbers being consecutive numbers.

31. (original) The graphics accelerator as defined by claim 29 the storing means comprises:

means for consecutively storing each consecutive data block of the one dimensional texture map in the located memory locations.

32. (currently amended) A computer program product for use on a computer system for storing a texture map in a single linear texture memory of a graphics accelerator, the computer program product comprising a computer usable medium having computer readable program code thereon, the computer readable program code including

program code for determining the dimension of the texture map;

program code for converting the texture map to a one dimensional texture map if the dimension of the texture map is determined to be more than one dimensional, the one dimensional texture map having a first number of consecutive data blocks;

program code for locating a second number of consecutive memory locations in the texture memory, the first number being equal to the second number; an

program code for storing the one dimensional texture map in the located memory locations in the single texture memory.

33. (original) The computer program product as defined by claim 32 wherein the texture map is two dimensional, the program code for converting comprising:

program code for defining a plurality of data blocks within the texture map; and

program code for assigning a sequence number to each of the data blocks, the sequence numbers being consecutive numbers.

34. (original) The computer program product as defined by claim 32 wherein the program code for storing comprises

program code for consecutively storing each consecutive data block of the one dimensional texture map in the located memory locations.

35. (currently amended) A data structure for storing data relating to a texture map, the texture map having an associated dimension and being stored at a given location in a single memory device, the data structure comprising

a location field identifying the given location in the memory device; and

a dimension field identifying the dimension of the texture map.

36. (original) The data structure as defined by claim 35 wherein the texture map comprises a set of mipmaps, further wherein the location field includes a plurality of subfields, each subfield identifying the location of one mipmap in the set of mipmaps.

37. (currently amended) The data structure as defined by claim 35 wherein the texture map spans a plurality of addresses in the single memory device, the location field identifying the plurality of addresses.

38. (currently amended) The data structure as defined by claim 35 wherein the data structure is stored in the single memory device, the single memory device being texture memory.